



BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

WORK INTEGRATED LEARNING PROGRAMMES

Second Semester 2025-2026

Digital

Part A: Content Design

Course Title	ML SYSTEM OPTIMIZATION
Course No(s)	AIML CLZG516
Credit Units	5
Credit model	
Content Authors	Madhusudhanan B
Lead Instructor	Madhusudhanan B
Version	1.0
Date	21-02-2026

Course Descriptions

Equip students with comprehensive knowledge of ML system optimization through exploration of computational performance, scalability on modern hardware architectures (multi-core CPUs, GPUs, clusters), and trade-offs between performance and model quality.

Course Objectives

No	Course Objective
C01	Develop a comprehensive understanding of machine learning system optimization, including computational efficiency, memory utilization, and algorithmic performance trade-offs.
C02	Examine how machine learning algorithms scale across multi-core CPUs, GPUs, and distributed cluster environments, and evaluate their performance characteristics on each platform.
C03	Design and implement machine learning workflows that leverage parallelism (data/model parallelism) and distributed computing frameworks to improve training throughput and system efficiency.

C04	Critically assess the balance between computational performance (speed, memory, scalability) and model quality (accuracy, convergence stability, generalization) when optimizing ML systems.
------------	--

Reference Book(s) & other resources

R1	Computer Architecture: A Quantitative Approach, Hennessy, J.L. & Patterson, D.A. (6th ed., Morgan Kaufmann, 2017)
R2	Distributed Machine Learning Patterns, Yuan Tang (Manning, 2023)
R3	
R4	
R5	
R6	
R7	
R8	

Learning Objectives: Students will be able to

L01	Design and implement parallel/distributed ML algorithms leveraging modern platforms
L02	Optimize ML models for deployment on resource-constrained devices
L03	Evaluate system-level performance trade-offs in ML pipelines

Content Structure

Modules

M1	Foundations	Performance Metrics & Complexity Analysis: Time/space complexity, throughput, latency, speedup (Amdahl's Law) Parallelization Paradigms: Data-level, task-level, and algorithm-specific parallelism Modern Hardware Architectures: Multi-core CPUs, GPGPUs, GPU/CPU clusters, resource management
M2	Distributed ML algorithms	Parallel ML Algorithms: CNN, Gradient Descent/SGD, SVM, k-Means, kNN, Decision Trees/Random Forests Distributed Training Strategies:

		Data parallelism, model parallelism, pipeline parallelism, gradient checkpointing, mixed precision training
M3	Scale-Out Systems	ML Platforms & Frameworks: Parameter Server Model, Spark MLlib, TensorFlow/PyTorch distributed training Distributed Deep Learning: Decentralized SGD, All-reduce, Asynchronous parallelism, Large-scale neural networks Federated Learning: Privacy-preserving distributed training, communication efficiency, hardware utilization
M4	Constrained Systems	Model Optimization Techniques: Model compression, pruning (structured/unstructured), quantization (post-training/quantization-aware) Edge Deployment: TinyML, TensorFlow Lite, ONNX Runtime, knowledge distillation Performance Trade-offs: Accuracy vs. latency, model size vs. throughput, energy consumption optimization

Part B: Learning Plan

Academic Term	Second Semester 2025-2026
Course Title	ML SYSTEM OPTIMIZATION
Course No	AIML CLZG516
Lead Instructor	Madhusudhanan B

Session Plan:

Session	Topics	Notes
M1	Foundations	
1	ML and DL: 1. Performance: a. Metrics: Time Complexity of Algorithms and Running Time; b. Scaling and Tuning of Performance 2. Environments: a. Training vs. Deployment b. Range of Systems: Distributed and Cloud, Embedded and Mobile.	• <i>No reading required;</i> • <i>Broad understanding required: of Algorithmic Complexity, and Performance metrics like Throughput and Response Time</i>
2	Parallel and Distributed Algorithms: 1. Systems and Performance; 2. Speedup – Approaches and Issues; 3. Data Parallelism vs. Task Parallelism vs. Request Parallelism. 4. Scale-out Clusters – Cost of communication and impact on Speedup	• <i>No reading required</i> • <i>Desired understanding: Speedup: Amdahl's Law, Scale-up vs. Scale-out</i>
3	1. Parallel Programming Models 2. Map-Reduce Pattern 3. Task-Parallel and Request-Parallel Parallelization of ML Algorithms - Examples 1. Multicore & GPGPU Parallelism - SIMD vs. MIMD execution - Memory hierarchies and bandwidth considerations 2. Distributed Clusters - CPU-only vs. GPU-accelerated clusters - Data sharding (horizontal vs. vertical) - Parameter-server vs. all-reduce paradigms	• <i>No reading required</i> • <i>Desired understanding: Parallel and Multi-core Processing</i>
M2 Distributed ML		
4	1. Introduction to problem decomposition 2. Parallelization of ML Algorithms	• <i>Prior Knowledge: ML algorithms</i>

	a. Ensemble Methods b. XGBoost in CPU c. K-Means Decision Trees & Random Forests - Distributed implementations (Spark-ML, XGBoost's distributed GPU mode) Support Vector Machines (SVM) - Parallelizing kernel computations (block-wise, map-reduce style)	• Reading: Specific Distributed ML algorithms
5	1. Distributed Algorithms – Communication Overhead Distributed ML Algorithms – Example: k-Means - Model Parallelism k-Nearest Neighbors (k-NN) - Data partitioning & locality-sensitive hashing - Distributed indexing (KD-tree, Ball-tree) vs. brute-force map-reduce - Trade-offs: accuracy vs. latency in approximate k-NN Gradient Descent & SGD - Minibatch vs. synchronous vs. asynchronous SGD	•
6	Clusters Hadoop and Spark k-Means Clustering - Lloyd's algorithm in MapReduce (each worker computes local centroids) - Mini-batch k-means & streaming variants - Convergence criteria and communication reduction (compressing updates) Convolutional Neural Networks (CNNs) - Data vs. model parallelism (layer splitting, tensor slicing) - Mixed-precision & pipeline parallelism - Distributed training frameworks (Distributed TensorFlow)	•
7	Distributed Training Strategies: Data parallelism, model parallelism, pipeline parallelism, gradient checkpointing, mixed precision training	•
M3. Scale-out Systems		
8	ML Platforms & Frameworks: Implementation Issues • The Parameter Server Model Stochastic Gradient Descent TensorFlow Architecture - Master, Worker, Parameter Server roles - Graph mode vs. eager execution	•

	- tf.distribute strategies (Mirrored, MultiWorker, ParameterServer) Performance Improvement & Trade-offs - Batch vs. micro-batch sizing - Model compression & quantization - I/O and shuffling overhead	
9	Distributed Deep Learning: Decentralized SGD, All-reduce, Asynchronous parallelism Towards Optimizing Neural Nets - System Optimization Techniques - Program Optimization Techniques - Matrix Multiplication All-reduce Paradigm - Ring vs. tree-based reduction - Bandwidth and latency considerations Asynchronous Parallelism - Hogwild! and parameter staleness - Impact on convergence and variance Local SGD - Periodic model averaging - Communication-computation trade-off	•
10	Locality-aware Programs, Massive Multi-threading Introduction to GPGPUs Model Parallelism - Layer slicing vs. tensor splitting - Cross-device communication patterns Pipeline Parallelism - Micro-batch pipelining - Bubble overhead and scheduling Scalability Case Studies - GPT-class models on GPU clusters - Memory-efficient checkpointing	•
11	GPU Architecture GPU Threading Model Case Study : Matrix Addition Multi-threaded Matrix Multiplication Federated Learning Basics - Data partitioning (non-IID) and privacy goals - Secure aggregation primitives Emerging Architectures - Decentralized FL (gossip, blockchain)	•
	M4. Constrained Systems	
12	CNN (GPU implementation) Model Compression Techniques	• <i>Prior Knowledge: Deep NNs</i>

	• Quantization • Network pruning • Low-rank factorization Overview of Model Compression - Motivation: storage, memory, bandwidth savings - Lossy vs. lossless compression Weight Sharing & Encoding - HashNet, weight clustering - Huffman and arithmetic coding Knowledge Distillation - Teacher–student paradigms - Temperature, soft labels, and cost functions	• Reading: Papers on Quantization, Model Compression
13	Model Compression Techniques – continued Knowledge distillation Edge Intelligence – Case Studies Quantization Fundamentals - Uniform vs. non-uniform quantization - Post-training quantization vs. quant-aware training Reduced-Precision Arithmetic - 16-bit, 8-bit, down to binary/ternary networks - Impact on forward/backward passes Hardware Support - CPU vector units, TPU bfloat16, MCU DSP blocks	•
14	CNN (GPU implementation) Federated Learning 1. Pruning Methods - Unstructured (magnitude-based) vs. structured (filter/channel) pruning - Iterative vs. one-shot pruning 2. Evaluation Metrics - Sparsity ratio, FLOPs reduction, accuracy drop 3. Deep Compression Pipeline - Combining pruning + quantization + encoding - Sparse matrix storage formats	• <i>Prior Knowledge: Deep NNs,</i> • Reading: Papers on Quantization, Pruning, and Compression Studies
15	Federated Learning(Contd) The Optimization Landscape: • Generic Approaches vs. Specific Approaches • Algorithm-Specific vs. Target-Platform Specific • ML Frameworks and Libraries • Hardware and Compilers Small Language Models - Frameworks: TFLite Micro	Reading: Papers on TensorFlow Lite Programming, Evaluation of Compression Techniques
16	Energy Constraints – Adapting Algorithms for Constrained Devices; Assessing the tradeoffs - Accuracy of prediction, Model Size, Throughput, Response Time, Energy Consumption	

Experiential Learning Components:

Describe objective, outcome of Experiential Learning Component and the lab infrastructure needed (virtual, remote, open source etc...) number of lab exercises needed, etc.

1. Lab work: 0
2. Project work: 0
3. Case Study: 4 Webinars
4. Simulation: 0
5. Work Integrated Learning Assignment- 1/2 Assignments
6. Design work/ Field work: 0

Objective of Experiential Learning Component:**Scope of Experiential Learning Component:**

Programming language - Python

Tools and libraries:

Lab Infrastructure:

Virtual Lab

Detailed Plan for Lab work

Lab No.	Lab Objective	Lab Sheet Access URL	Session Reference
1			
2			
3			
4			
5			
6			
7			
8			
9			

Evaluation Scheme:

Legend: EC = Evaluation Component; AN = After Noon Session; FN = Fore Noon Session

No	Name	Type	Duration	Weight	Day, Date, Session, Time
EC-1(a)	Quizzes	Online		10%	
EC-1(b)	Assignments	Take Home		20%	
EC-2	Mid-Semester Test	Closed Book		30%	
EC-3	Comprehensive Exam	Open Book		40%	

Note:

Syllabus for Mid-Semester Test (Closed Book): Topics in Session Nos. 1 to 8

Syllabus for Comprehensive Exam (Open Book): All topics (Session Nos. 1 to 16)

Important links and information:

Elearn portal: <https://elearn.bits-pilani.ac.in> or Canvas

Students are expected to visit the Elearn portal on a regular basis and stay up to date with the latest announcements and deadlines.

Contact sessions: Students should attend the online lectures as per the schedule provided on the Elearn portal.

Evaluation Guidelines:

- 1 EC-1 consists of two Quizzes. Students will attempt them through the course pages on the Elearn portal. Announcements will be made on the portal, in a timely manner.
- 2 EC-2 consists of either one or two Assignments. Students will attempt them through the course pages on the Elearn portal. Announcements will be made on the portal, in a timely manner.
- 3 For Closed Book tests: No books or reference material of any kind will be permitted.
- 4 For Open Book exams: Use of books and any printed / written reference material (filed or bound) is permitted. However, loose sheets of paper will not be allowed. Use of calculators is permitted in all exams. Laptops/Mobiles of any kind are not allowed. Exchange of any material is not allowed.
- 5 If a student is unable to appear for the Regular Test/Exam due to genuine exigencies, the student should follow the procedure to apply for the Make-Up Test/Exam which will be made available on the Elearn portal. The Make-Up Test/Exam will be conducted only at selected exam centres on the dates to be announced later.

It shall be the responsibility of the individual student to be regular in maintaining the self-study schedule as given in the course hand-out, attend the online lectures, and take all the prescribed evaluation components such as Assignment/Quiz, Mid-Semester Test and Comprehensive Exam according to the evaluation scheme provided in the hand-out.